NASA Unified WRF (NU-WRF) Version 6-3.4.1 User Guide

Document Revision: 10
12 August 2013

1. Overview

The Weather Research and Forecasting (WRF) model is a non-hydrostatic numerical weather prediction system designed for portability, efficiency, and applications in both research and operations. WRF can be used for a wide range of configurations ranging from idealized large-eddy simulations to global modeling, with an emphasis on horizontal grid sizes in the range of 1–13 km. When simulating regional weather and climate, WRF is typically run within a domain covering several thousand kilometers using interactive nesting techniques with multiple grid refinements. Two dynamic solvers are available in WRF: The Advanced Research WRF (ARW, maintained by NCAR) and the Nonhydrostatic Mesoscale Model (NMM, maintained by NCEP and DTC). See http://www.mmm.ucar.edu/wrf/users/docs/arw_v3.pdf for technical details on the ARW solver.

At NASA, WRF-ARW is used to study a variety of phenomenon such as high impact weather and aerosol effects. Several parameterizations developed by NASA scientists have been implemented in WRF to better represent and simulate cloud-aerosol-precipitation-land surface processes. NASA Unified WRF (NU-WRF) combines these improvements in order to: (1) facilitate better use of WRF in scientific research; (2) reduce redundancy in major WRF development at NASA; (3) prolong the serviceable life span of WRF; and (4) allow better use of NASA high-resolution satellite data for research (short term climate and weather).

NU-WRF Version 6-3.4.1 was released in **August 2013**. The version number "6-3.4.1" (or more simply, "Version 6") designates the sixth official release of NU-WRF based upon the WRF 3.4.1 code base. The NU-WRF software project consists of multiple software packages and enhancements built around the WRF modeling system, use cases with input data sets, and documentation describing the software and software processes.

2. Current Components

NU-WRF software consists of the following software components (new features highlighted

in blue):

- WRF-ARW 3.4.1 (including WRF/Chem 3.4.1) with NASA enhancements:
    - Updated Goddard microphysics
    - Updated Goddard radiation (longwave and shortwave)
    - On-line coupling between GOCART aerosols and Goddard radiation
    - On-line coupling between GOCART aerosols and Goddard microphysics
    - On-line coupling with NASA Land Information System (LIS) 6.1
    - Estimation of SOA from biogenic terpene emissions
    - Linking of MEGAN2 biogenic emissions scheme to GOCART related chemistry
    - Linking of RADM2 chemistry to GOCARTRADM2 option;
    - Linking various optical property schemes to GOCART related chemistry
    - Linking GOCART dry deposition scheme to GOCART related chemistry
    - “Sanity check” module to identify unphysical values passed between coupled physics parameterizations
    - Severe weather diagnostics (lightning flash rate, etc.)
    - Added support for 72-level GMI “background fields” for GOCART
    - User-tunable constants for GOCART dust emissions
    - Use of LIS porosity (from Noah 2.7.1, Noah 3.1, Noah 3.2, and CLM2 options) in GOCART dust emissions
    - Modifications to REAL to process LIS netCDF history files for input to WRF
    - Support for the UMD land use scheme when running LIS coupled with WRF
- WPS 3.4.1 (Standard WRF pre-processors for terrain, land-use, and initial/lateral boundary conditions) with NASA enhancements:
    - Fixed processing ERA-Interim sea surface temperatures
- UPP 2.0 (Unified Post Processor/GRIB generator from NCEP) with NASA enhancements:
    - Severe weather diagnostics
- MET 4.0 (verification package)
- ARWpost 3.1 (WRF post-processor/GrADS and Vis5D file generator)
- RIP 4.6.3 (WRF post-processor/NCAR Graphics plotting software) with NASA enhancement:
    - Bug fix to allow processing of WRF/Chem netCDF output files
- NASA LVT (LIS Verification Toolkit)
- NASA Goddard SDSU V3.0 (simulates satellite measurements from WRF gridded data)
- prep_chem_sources 1.2 (emissions preprocessor) with NASA additions:
    - Supports GFEDV3 fire emissions
    - Uses WPS map projection library

    - Fixed processing of GOCART “background fields” from GMI
    - Added support for new 72-level GMI files for GOCART “background fields”
    - Outputs additional GrADS binary and control files, and ASCII map projection files for plotting
- NASA plot_chem (simple NCAR Graphics plotting program for output from prep_chem_sources developed at NASA)
- NASA GEOS2WRF 2.0 (for NASA GEOS-5 global model output)
    - General front end converter (`geos2wps`) to extract variables from HDF-EOS, netCDF3, or netCDF4 files and convert to WPS intermediate format
    - Utility to calculate GEOS-5 surface terrain field (`createSOILHGT`)
    - Utility to calculate geopotential heights on GEOS-5 model levels (`createHGT`)
    - Utility to create land sea table from GEOS-5 (`createLANDSEA`)
    - Utility to calculate mid-level pressures in GEOS-5 model coordinates (`createPRESSURE`)
    - Utility to calculate relative humidity from GEOS-5 (`createRH`)
    - Utility to extrapolate GEOS-5 data to underground isobaric levels (`extrapIsobaric`)
    - Utility to split WPS intermediate file into individual files containing single slab (`splitWPS`)
    - Source code in new `utils/geos2wrf_2` directory
- NASA MERRA2WRF 2.0 (for NASA MERRA reanalysis output)
    - Processes HDF4, HDF-EOS and netCDF3 files from NASA GES DISC web page; source code in new `utils/geos2wrf_2` directory
- NASA GOCART2WRF 2.0 (for NASA GOCART global aerosol model output)
    - Processes netCDF4 files from GEOS-5; source code in `utils/gocart2wrf_2` directory
- NASA SST2WRF (for sea surface temperature analyses from Remote Sensing Systems)
- NASA LISConfig (customizes LIS domain to match that of WRF)
    - Now processes output from `geogrid.exe`
- NASA unified build system for compiling software components:
    - Supports building NU-WRF software on both Discover and Pleiades with Intel compilers and Intel MPI

3. New Vendor-Specific Updates

- WRF-ARW 3.4.1 (from WRF-ARW web page):

- Reset cudt = 0 for BMJ, all SAS, and Tiedtke cumulus schemes to ensure correct time-step rain when adaptive time step is used
- Bug fixes for cumulus calls with adaptive time step when OpenMP is used
- Moved glacier ice treatment out of Noah code into separate subroutine
- Bug fixes to Noah MP LSM; changed namelist options to single value (non-domain dependent)
- Added iz0tind option and modified isftcflx options for MYNN surface layer
- Some tuning of MYNN PBL scheme, and added option to output diagnostics (bl_mynn_tkebudget=1)
- Bug fix to YSU PBL scheme for stable surface conditions and consistency with thermal roughness length -- may improve surface wind forecast at night
- Bug fix to QNSE/EDMF (day time) PBL scheme
- Activated seaice function in SSiB LSM scheme
- Changed Cd and Ck formulas for isftcflx=2 (Garratt-Donelan tropical cyclone drag coefficients)
- Changed graupel intercept parameter and other changes in Thompson microphysics
- Bug fixes to GD and G3 cumulus schemes when cu_diag is turned on
- Add feedback to surface diagnostic fields (T2, TH2, Q2, U10, V10, and PSFC)
- Fixed memory allocation bug with Shutts stochastic kinetic-energy backscatter scheme
- Variable description changed for ‘climate diagnostic’ output variables
- Updated input sounding for idealized two-fire example
- Bug fix to ndown.exe to use default hypsometric_opt=2 height calculation option
- Bug fix to correct water vapor mixing ratio at top half-level coming out of real.exe
- Bug fix to force output of history dump after restart (using write_hist_at_0h_rst=.true.)
- Experimental netCDF4 compression and chunking option. Can reduce file size up to 45%, but “it is a bit clumsy to use this option in this release, unless you’re running on bluefire at NCAR.” **[NOTE: This option is disabled in NU-WRF.]**
- Experimental in-code radar reflectivity calculations for WSM5/6, WDM5/6, Thompson, Morrison, Goddard, and Lin microphysics (including fix for spherical snow assumption with Thompson scheme)
- Experimental pressure level diagnostics controlled by new &diag namelist record

- WRF/Chem 3.4.1 (using 'svn diff'):
    - Added NO2 to plume rise model
    - Added MEGAN v2.04 species conversion to RACM_SOA_VBD_KPP species
    - Added RACM_SOA_VBS_KPP to optical driver, module_bioemi_megan2, and dry_dep_driver
    - Added 4-bin volcanic ash (CHEM_VOLC_4BIN) option to emissions driver
    - Added CHEM_VOLC_4BIN option to chem_driver and module_input_chem_data
    - Fixed array dimension in module_aerosols_sorgam
    - Bug fix in wet deposition code
    - Bug fixes to volcanic ash settling code
    - Switched out Price-Rind resolution dependency for area ratio adjustment in lightning parameterization
    - Bug fix to `convert_emiss.exe` to read in OL2 emissions
    - Bug fix to module_add_emiss_burn to include NO2
- WPS 3.4.1 (from WRF-ARW web page):
    - Fixed `Vtable.CFSR_sfc_flx06` input file to reflect reading in specific humidity instead of relative humidity
    - Bug fix to correctly process GFS GRIB data after 22 May 2012
    - Corrected `configure` script to avoid erroneously printing 'Your Fortran + NETCDF did not run successfully' messages
    - Updated USGS land cover over Iceland for more normal glacier coverage
- RIP 4.6.3 (based on 'svn diff'):
    - No updates in this NU-WRF release
- UPP 2.0 (from DTC web page)
    - Various unused subroutines removed to support merge of DTC/NCEP versions of post
    - `MISCLN.f` and `CALHEL.f` modified to fix bug that caused post to output 1-km helicity with values of 3-km helicity when only 3-km helicity was requested
    - Changed names for all subroutines involving precipitation type derivation (`SURFCE.f` and `CALWXT*.f`)
    - Modified `MDL2P.f` to fix bug that output isobaric dew point temperature when users requested Haines Index, and to correct 3rd index to O3 array to avoid out of bounds error
    - `CLDRAD.f` and `AVIATION.f` modified to properly bit map out ceiling when instantaneous total cloud fraction is unavailable in model output
    - Bug fix in `MDLFLD.f` for computing transport wind and ventilation rate

    - Modified `RQSTFLD.f` to add new variables and correct GRIB identifiers for NCAR in-flight icing
    - Modified `FDLVL.f` and `MISCLN.f` to add flight level icing field
    - Changed `CTLBLK.f` to add 16th flight level at 7010 m
    - Properly zero out convective precip rate for ARW when unavailable in model output
    - Minor change to `AVIATION.f` to improve if statement structure
    - Bug fixes to `CALRAD_WCLOUD_newcrtm.f` to correct kind types of arguments used for calling snfrac, and to specify main land type to snow (instead of land) at grid points with permanent ice/snow even after snow melts in the summer
    - Changed `CALCAPE.f` to fix out-of-bounds array reference for temperature
    - Changed gust calculation using Richardson number derived PBL height of all models
    - `INITPOST_BIN_MPIIO.f` changed to read CWM properly
    - Modified `AVIATION.f` to fill in the gap at 0 meridian for global aviation products, and to fix out-of-bounds array error when computing CAT for an A-grid
    - Bug fixes to `SNFRAC.f` to computer snow fractions for new IGBP veg type
    - Changed `SURFCE.f` to use index 116 for soil temperature for all LSMs
    - Modified `CLDRAD.f` to fix out-of-bounds referencing for do loop 373, and to correct boundary condition when computing zcldtop
    - Added serial support for new MPI routines called to wrfmpi_stubs
    - Remove dependency in typeset for `scripts/run_unipost` (not cross platform compliant)
    - Convert `scripts/run_unipost*` and `src/unipost/gribit.F` to return 2 or 3 digit forecast hour in output filename
    - Removed serial binary support
    - Changed `INITPOST.f` to rename surface runoff variable to SSROFF, and to set SLDPTH based on model
    - Add support of CRTM ssmis_f17 coefficient files
    - Add conditional around print statement in `MISCLN.f` to avoid failure due to unallocated structure
    - Upgraded SIGIO library to match EMC version
    - Flipped ARW data so Z dimension to increase with pressure data
    - Changed tile definition to allow MPI runs on Linux machines
- MET 4.0 patches (from DTC web page):
    - Fixed the `trmm2nc.R` script
    - Fixed a PCP-Combine bug for the -add option

    - Fixed a PGI compilation problem
    - Fixed a GRIB2 library bug that caused gridding problems
    - Fixed a bug that caused a mixup in GRIB2 forecast data values when multiple forecast variables are verified
    - Fixed Stat-Analysis bug causing mixup in GRIB2 forecast data values when reading MPR lines with a config file job
    - Fixed bug causing segmentation fault when config file string list contains only an empty string
    - Adjusted order of system include files to fix a problem when MET is compiled with PGI compiler family
    - Fixed bug affecting GRIB2 support when using PGI compiler
    - Added gslcblas library to `Makefile` link for `madis2nc`, `plot_point_obs` and `gen_poly_mask`
- ARWpost 3.1 (converts WRF output to GrADS format)
    - No updates in this NU-WRF release

4. NASA Updates New to Current Version

- NASA Land Information System (LIS)
    - Added uncoupled LIS as a new executable build target. Users should use `./build.sh lis`.
    - Improved processing of LIS netCDF history dumps by REAL, replacing certain land surface fields provided by metgrid.exe. Users must use the `lisreal` build option to activate, and also include `netcdf4` if processing files from LIS7. If the `lisreal` option is used, real.exe will expect LIS netCDF history files and will fail if they are not present.
    - Disabled changes in WRF land/sea table due to sea ice fraction. Users must use the `lisreal` build option to activate. This should preserve agreement between WRF and LIS understanding of land locations.
    - Modified LANDUSE.TBL to support UMD land use entries when running coupled WRF/LIS
    - Bug fix: LIS now uses correct time step for each domain when running in nested mode
    - Added WRFout plugin to use WRF netCDF files for atmospheric forcing of uncoupled LIS
    - Added terrain adjustment option for climatological deep soil temperature in LIS (same as that used in REAL)
    - Added dynamic deep soil adjustment option as function of lagged skin temperature (based on that used in WRF)

    - WRF now passes deep soil temperature to LIS in coupled mode
    - Bug fix: Can now write deep soil temperature to LIS netCDF file
    - Now writes snowt as instantaneous value to LIS netCDF file
    - Now writes albedo and max snow albedo to LIS netCDF file
    - Bug fix to writing netCDF output header
    - Added WPS reader plugin
    - Reorganized LIS_histDataMod.F90 to have variables closer to ALMA order and grouping, and updated test cases accordingly
    - Bug fixes to MERRA forcing pluging
    - Corrected output of SMLiqFrac
    - Updated lis.config documentation
    - Bug fix: Upscaling code did not properly check mask and neighbors
    - Bug fix: Remove extraneous ssdev argument from call to LIS_readObsAttributes
- NASA Goddard radiation
    - No updates
- NASA Goddard microphysics
    - Added rainncv_sepa and rainnc_sepa variables
- WRF/Chem/GOCART
    - No updates
- NASA Goddard SDSU
    - Added new routines for calculating particle size distribution variables
    - Migrated physical constants to new module
    - New code to calculate cosine of incidence angle
    - New module for radar equation for down-looking pencil beam radar
    - New module for calculating optical properties of GOCART aerosols
    - Added scan simulator for A-Train satellite
    - New module to estimate antenna gain function of satellite for non-scan simulation
    - Now outputs satellite orbit information
    - New lookup tables
    - Changed QRUN directory names (`QRUN_DISCOVER` is now `QRUN_NCCS`, `QRUN_PLEIADES` is now `QRUN_NAS`)
- NASA Data Utilities
    - No updates
- NASA build system
    - The `lisreal` option now compiles REAL such that it processes LIS netCDF file(s) and returns an error if no files exist. Also, the `lisreal` option will compile WRF such that the land/sea table is frozen during the run.

- Added `netcdf4` option to link WRFV3 executables against netCDF4 library (including HDF5, ZLIB, and SZIP). This allows real.exe to read in and process netCDF4 files from LIS 7. **[NOTE: The build system still does not support building WRFV3 executables to generate netCDF4 output.]**
- Template makefiles and/or compile options are now specified in the build system cfg files. This should make it easier to port NU-WRF to new computer systems, new compilers, and/or new MPI implementations.
- Reorganized sanity check code to ensure WRFV3, WPS, and UPP are all compiled either with or without MPI (no mixing and matching)
- Changed method for identifying absolute paths of files (original version relied on behavior specific to GNU `readlink`)
- Changed QRUN directory names for GSDSU
- Updated `discover.cfg` to specify third-party libraries compiled on SLES11.1

5. Using the Software

a. Acquiring the Source Code

Reference the *NU-WRF Source Code and Data* documentation at: https://modelingguru.nasa.gov/docs/DOC-1834 for download instructions. **Note that the release of NU-WRF software is subject to NASA legal review and requires users to sign a Software Usage Agreement. Please contact Christa Peters-Lidard (christa.d.peters-lidard@nasa.gov) for more information.**

b. Required compilers

Users must have Fortran 90 and C compilers (and C++ if compiling MET). Currently NU-WRF uses the Intel Fortran, C, and C++ compilers version 12.1.0 on the Discover and Pleiades supercomputers (at NASA Goddard Space Flight Center and NASA Ames Research Center, respectively).

c. External Libraries

The following external libraries are used to build the entire NU-WRF software package:

- BUFRLIB 10.0.1
- ESMF 3.1.0rp3 (compiled with and without MPI support)
- GSL 1.15
- G2CLIB 1.2.3

- Flex (binary and library)
- HDF4 4.2.6
- HDF5 1.8.7
- JasPer 1.900.1
- JPEG 6b
- Libpng 1.2.46
- MPI (currently Intel MPI 4.0)
- NETCDF 4.1.3 (built with and without HDF5 compression support)
- NCAR Graphics 6.0.0
- SZIP 2.1
- YACC (binary)
- ZLIB 1.2.5

These libraries are installed on Discover in `/usr/local/other/SLES11.1`, except for YACC (found at `/usr/bin/yacc`), FLEX (binary in `/usr/bin`, library in `/usr/lib64`), and Intel MPI (load the mpi/impi-4.0.1.007-beta module). On Pleiades, these are found in `/nobackupp1/nuwrf/lib/SLES11`, except for YACC (found in `/usr/bin/yacc`), FLEX (binary in `/usr/bin`, library in `/usr/lib64`), and Intel MPI (load the mpi-intel/4.0.2.003 module). Aside from YACC, FLEX, and Intel MPI, **all external libraries must be built using the same compilers as NU-WRF itself.**

Users can experiment with alternative MPI implementations (e.g., MVAPICH2) and newer versions of the above libraries. **NOTE: Libpng 1.5.* will not link with WPS.**

d. Building the Source Code

The NU-WRF modeling system is composed of a number of software packages, each of them with their own separate build system. To make it easier for the user to create desired executables and to more easily resolve dependencies between packages, NU-WRF Version 6 includes a set of high-level "wrapper" scripts for building. With these scripts, the user can build executables using a single script called **`build.sh`** located in the top-level directory. This script accepts three types of command-line arguments:

- *Configuration*. The **`--config`** flag followed by the name of the configuration file specifying system specific environment variables (e.g., the path to the netCDF library). Currently two files are included in the top-level directory of the distribution: **`discover.cfg`** and **`pleiades.cfg`**. Users may develop their own configuration file to customize their settings. If no choice is given for the configuration file, the script will default to either **`discover.cfg`** or **`pleiades.cfg`** based on the local environment.

- *Options.* The user may specify **`lisreal`**, **`cleanfirst`**, **`debug`**, **`netcdf4`** and/or **`nest=1`**, **`nest=2`**, or **`nest=3`**. The **`lisreal`** option indicates that if WRF is built, then the resulting `real.exe` executable will generate initial conditions compatible with LIS, **and will expect LIS netCDF history files to process**. Also, wrf.exe will be compiled such that the land/sea mask **will remain constant** during the integration (regardless of sea ice). The **`cleanfirst`** option will cause the build system to "clean" a target (delete object files and static libraries) before building it. The **`debug`** option forces the WRF or WPS build systems to use alternative compiling flags set in the configuration file (e.g., for disabling optimization and turning on run-time array bounds checking). The **`nest=1`**, **`nest=2`**, or **`nest=3`** options specify compiling with basic nesting, preset-moves nesting, or vortex-following nesting (**`nest=1`** is assumed by default). The **`netcdf4`** option will link the WRFV3 programs against the NetCDF4 library (with HDF5 compression), allowing for processing of LIS7 netCDF4 files. (Note that NU-WRF does not yet support compressed netCDF4 output.)
- *Full Build Targets*. The user can specify any of these to build the entire system:
  - **`all`** (Build all executables w/o chemistry)
  - **`allchem`** (Build all executables w/ chemistry but w/o Kinetic Pre-Processor)
  - **`allkpp`** (Build all executables w/ chemistry and w/ Kinetic Pre-Processor)
- *Clean Targets.* The user can use these to remove build products.
  - **`allclean`** (Delete all executables, object files, and static libraries)
- *Specific Component Targets*. The user can specify one or more of these targets for the build system during a build or regular clean step:
  - **`arwpost`** (Builds executables in `ARWpost` directory)
  - **`chem`** (Builds executables in `WRFV3` directory w/ chemistry but w/o Kinetic Pre-Processor, including `conv_emiss`)
  - **`geos2wrf`** (Builds executables in `utils/geos2wrf_2` directory)
  - **`gocart2wrf`** (Builds executables in `utils/gocart2wrf_2` directory)
  - **`kpp`** (Builds executables in `WRFV3` directory w/ chemistry and w/ Kinetic Pre-Processor, including `conv_emiss`)
  - **`lis`** (Builds uncoupled LIS executable in `WRFV3/lis/make` directory)
  - **`lisconfig`** (Builds executables in `utils/lisconfig` directory)
  - **`lvt`** (Builds executables in `LVT` directory)
  - **`merra2wrf`** (Builds executables in `utils/geos2wrf_2` directory)
  - **`met`** (Builds executables in `MET` directory)
  - **`plot_chem`** (Builds executables in `utils/plot_chem` directory)
  - **`prep_chem_sources`** (Builds executables in

`utils/prep_chem_sources` directory)
- **`rip`** (Builds executables in `RIP4` directory)
- **`sdsu`** (Builds executables in `GSDSU` directory)
- **`sst2wrf`** (Builds executables in `utils/sst2wrf` directory)
- **`upp`** (Builds executables in `UPP` directory)
- **`wps`** (Builds executables in `WPS` directory)
- **`wrf`** (Builds executables in `WRFV3` directory w/o chemistry, excluding conv_emiss)

In practice, the **`build.sh`** script may (re)build additional targets to resolve dependencies between programs.

The most straight-forward way to build the full system on Discover or Pleiades is to run the build script from the top-level folder:

```
./build.sh all
```

Or to build with chemistry enabled:

```
./build.sh allchem
```

To build a special version of `real.exe` to produce initial conditions compatible with LIS, the **`lisreal`** option must be included:

```
./build.sh lisreal all
```

And to fully clean the package, type:

```
./build.sh allclean
```

Users can also explicitly specify the configuration file, e.g.,

```
./build.sh --config discover.cfg lisreal all
```

Note that the build system will automatically select the **`discover.cfg`** (**`pleiades.cfg`**) if it detects the software is being built on Discover (Pleiades). The **`--config`** flag option is intended for future development and porting.

The user can selectively build packages by listing specific targets. For example, to build the WRF model without chemistry along with WPS and UPP:

```
./build.sh wrf wps upp
```

e. About the Build System

The top-level **`build.sh`** calls lower-level `build.sh` wrapper scripts located in each

package directory (WRFV3, WPS, etc.). Configuration settings are passed to the lower-level scripts via environment variables. Each lower-level script is customized to directly manage the component-specific build mechanism (e.g., the `configure` and `compile` scripts for WRFV3, `make` for `utils/geos2wrf`, etc.), and to inject the appropriate configuration settings into that build mechanism. For example, the `build.sh` for WPS will modify the `configure.wps` file generated by `configure` to update several library paths; the modified `configure.wps` is then used by the `compile` script.

Beginning in NU-WRF Version 6, the build system allows users to specify `configure` options for ARWpost, RIP4, UPP, WPS, and WRFV3, as well as template `Makefile` names for geos2wrf, gocart2wrf, GSDSU, lisconfig, LVT, MET, prep_chem_sources, and sst2wrf. Users can also specify whether or not to compile UPP, WPS, and WRFV3 with MPI. These options are stored in the build config file (currently `discover.cfg` and `pleiades.cfg`). This change should make it significantly easier to port NU-WRF to new compilers, MPI implementations, and/or operating systems; however, users will need to first create new makefile templates and/or modify default configure architecture files (see Appendix A).

One complication addressed by the build system is that two packages (WPS and UPP) are dependent on libraries and object files from WRFV3. To account for this, the build script has the following behavior:

- If both WPS and UPP are to be built, a check is made to ensure both are compiled with or without MPI. If inconsistent MPI selections are found for WPS and UPP, the build system will abort.
- If WPS and/or UPP are to be built, WRFV3 will be checked to ensure it was already built with a consistent MPI or non-MPI option. The WRFV3 will also be check to ensure the object files and static libraries required by WPS and/or UPP exist. If any of these checks fail, WRFV3 will be cleaned and automatically built with the appropriate MPI or non-MPI option. (Exception: If the config file has an inconsistent MPI selection for WRFV3, the build system will abort.) **This will occur even if the `wrf` target is not listed on the command line.**

An additional complication is specific to NU-WRF: The coupling of LIS to WRF requires linking WRFV3 to ESMF 3.1.0rp3 and to ZLIB. As a result, the `configure.wrf` is modified to link against these libraries. A similar modification is in place for UPP. (No modification is needed for WPS as long as WPS is compiled for GRIB2 support.)

6. Running NU-WRF

a. Background

Running NU-WRF largely follows the steps used with the community versions of the software. Users should be familiar with the on-line documentation of these versions:

- WRF-ARW User's Guide (includes WPS and ARWpost documentation): http://www.mmm.ucar.edu/wrf/users/docs/user_guide_V3/contents.html
- WRF/Chem User's Guide: http://ruc.noaa.gov/wrf/WG11/Users_guide.pdf
- LIS 6.1 User's Guide: https://modelingguru.nasa.gov/servlet/JiveServlet/downloadBody/2071-102-5-5529/usersguide.pdf
- UPP 2.0: http://www.dtcenter.org/wrf-nmm/users/docs/user_guide/V3/users_guide_nmm_chap7.pdf
- MET 4: http://www.dtcenter.org/met/users/docs/users_guide/MET_Users_Guide_v4.0.pdf
- RIP4 : http://www.mmm.ucar.edu/wrf/users/docs/ripug.htm

For a general guide to the WRF software, reference the *WRF on Discover* documentation at:
https://modelingguru.nasa.gov/docs/DOC-1671

b. Differences with WRF

If the NU-WRF `wrf` target is built with the `lisreal` option, then real.exe will expect a new namelist block in namelist.input:

```
&lis
 lis_landcover_type = 3,
 lis_filename = '200607140000.d01.lis7test_water.nc',
 /
```

Here lis_landcover_type specifies what land classification system was used with LIS (1 = USGS, 2 = MODIS, 3 = UMD), and lis_filename is an array of strings specifying the LIS netCDF history files to read in for processing (one for each WRF domain). **LIS files are required if the `lisreal` build option is used**.

Other special NU-WRF options exist in the namelist.input file and are used with WRF, REAL, WRF/Chem, and convert_emiss:

- If running coupled WRF/LIS with a LIS spinup that used the UMD land use

classification, set num_land_cat=14 in the &physics namelist
- To use the Goddard microphysics scheme, set mp_physics=7 in the &physics namelist
- To use the Goddard radiation scheme, set ra_lw_physics=5 and ra_sw_physics=5 in the &physics namelist
- To run LIS coupled with NU-WRF, set sf_surface_physics=5 in the &physics namelist; note that only the LIS NOAH and LIS CLM2 options are supported, and running with LIS requires an additional lis.config file
- To estimate SOA from biogenic terpene emissions, set bio_emiss_soa=1 in the &chem namelist
- To enable coupling between the Goddard microphysics and GOCART aerosols, set gsfcgce_gocart_coupling=1 in the &chem namelist
- To enable coupling between the Goddard radiation and GOCART aerosols, set gsfcrad_gocart_coupling=1 in the &chem namelist
- To process new 72-level GMI "background field" files for GOCART in convert_emiss, set new_gocart_bg_files=.true. in the &chem namelist
- To tune the GOCART dust emissions algorithm, set the new gocart_dustemiss_gwetthresh and gocart_dustemiss_resfactor variables in the &chem namelist (both are 0.2 by default), which should be set for each WRF domain. The former sets the soil wetness threshold below which dust emissions are possible; the latter is a multiplication factor to adjust the dust mixing ratio according to the domain grid resolution.

Users who run NU-WRF with LIS coupling must also have the following (see also the LIS User's Guide):

- A `lis.config` file, with:
    - "Running mode" set to 8 (indicating coupled mode)
    - Start and end date/times matching those in namelist.input
    - Numbers of processors along x and y set so that the product equals the total number of MPI processes being used
    - Appropriate entry for LIS restart file (entry varies by land surface model selection)
    - "Forcing variables list file" set to the path of the `forcing_variables.txt.wrflist` file in `WRFV3/lis/configs` directory
    - "Model output attributes file" set to the path of the `MODEL_OUTPUT_LIST.TBL` file
    - Appropriate settings for input land surface data files and `LS_PARAMETERS` files for selected land surface model

- A LIS restart file for the same grid and using the same land surface model
- The `forcing_variables.txt.wrflis` file
- An appropriate `MODEL_OUTPUT_LIST.TBL` file
- Appropriate input land surface files (terrain, albedo, soil type, etc.) as the same resolution(s) as the LIS grid(s)
- Appropriate `LS_PARAMETERS` files for the selected land surface model

c. Differences with prep_chem_sources

Users who run prep_chem_sources should be aware of the following new option in prep_chem_sources.inp:

- To process GFEDV3 data, set use_gfedv3=1 and gfedv3_data_dir='/path/to/files' in the &RP_INPUT namelist

d. Differences with UPP

Users who run UPP should be aware of several new variables that can be listed in `wrf_cntlr.parm`: 'TOTAL COLUMN GRAUPEL', 'MAX VERT INTEG GRAUP', 'LIGHTNING THREAT 1 ', 'LIGHTNING THREAT 2 ', 'LIGHTNING THREAT 3 ', 'MAX LTG THREAT 1 ', 'MAX LTG THREAT 2 ', 'MAX LTG THREAT 3 '

e. Miscellaneous notes

The programs developed at NASA (GSDSU, GEOS2WRF, GOCART2WRF, LISCONFIG, LVT, MERRA2WRF, PLOT_CHEM, and SST2WRF) have `README` files or user's guides included with their source code.

Sample batch scripts for several executables (e.g., `geogrid.exe`, `ungrib.exe`, ...) are included in the NU-WRF distribution in the scripts directory, and use a common `config.sh` script to specify common environment variables. These batch scripts are designed to run on Discover using the general_hi queue, and users will likely have to modify them to run on different computers. Also, some tweaks may be required to customize a script for a particular experiment (e.g., `run_ungrib.sh` may need changes to process GRIB files with a unique naming convention).

f. General steps

General steps to running NU-WRF programs:

1. Optionally plot WRF domains based on a `namelist.wps` file: Run

WPS/util/plotgrids.exe

2. Process static terrestrial data: Run **WPS/geogrid.exe**
3. Optionally run LIS in spin-up mode
    a. Customize a `lis.config` file to run LIS using same grid as geogrid, run **utils/lisconfig/bin/lisWrfDomain** (via **utils/lisconfig/scripts/lisWrfDomain.py**)
    b. Run **WRFV3/lis/make/LIS**
4. Process gridded time-dependent data:
    a. For GRIB data, run **WPS/ungrib.exe**
        i. To list contents of GRIB1 file, run **WPS/util/g1print.exe**
        ii. To list contents of GRIB2 file, run **WPS/util/g2print.exe**
        iii. For ECMWF GRIB files on sigma coordinates, also run **WPS/util/calc_ecmwf_p.exe**
        iv. For UKMET GRIB files, also run **WPS/util/height_ukmo.exe**
    b. For MERRA HDFEOS, HDF4 or netCDF data from NASA GES DISC, run **utils/geos2wrf_2/merra2wrf**
    c. For GEOS HDF4 data, run **utils/geos2wrf_2/geos2wps**
        i. If surface terrain is missing, also run **utils/geos2wrf_2/createSOILHGT**
        ii. If geopotential height is missing when processing GEOS-5 model level data, also run **utils/geos2wrf_2/createHGT**
        iii. If a land/sea mask is missing, also run **utils/geos2wrf_2/createLANDSEA**
        iv. If GEOS-5 mid-layer pressures are missing (when processing model-level data), also run **utils/geos2wrf_2/createPRESSURE**
        v. If relative humidity is missing, also run **utils/geos2wrf_2/createRH**
        vi. If processing GEOS-5 isobaric data, also run **utils/geos2wrf_2/extrapIsobaric** to provide values underground
        vii. The **utils/geos2wrf_2/splitWPS** can be used to split a WPS intermediate format file into smaller files that each contain only a single 2D slab of data (useful for subsetting data from `geos2wps` for input into one of the other GEOS2WRF utilities)
    d. For global sea surface temperatures from Remote Sensing Systems, run **utils/sst2wrf/bin/sst2wrf**
5. Optionally remove levels of data from files produced in step 3: Run **WPS/util/mod_levs.exe**

6. Optionally calculate daily mean surface temperature: run **WPS/util/avg_tsfc.exe**
7. Optionally display output from steps 3-5:
    a. For plots, run **WPS/util/plotfmt.exe**
    b. For text listing, run **WPS/util/rd_intermediate.exe**
8. Horizontally interpolate static and time dependent data from Steps 3-6: Run **WPS/metgrid.exe**
9. Vertically interpolate data from `metgrid.exe`, optionally adjust land surface fields with LIS data, and generate initial and lateral boundary condition files: Run **WRFV3/run/real.exe**
10. Optionally process emissions data for WRF/Chem:
    a. First, run **utils/prep_chem_sources/bin/prep_chem_sources_RADM_WRF_FIM.exe**
    b. Second, optionally run **utils/plot_chem/plot_chem** for NCAR Graphics images
    c. Finally, run **WRFV3/chem/convert_emiss.exe**
11. Optionally add GOCART aerosol data from GEOS-5 to initial and lateral boundary condition files: Run **utils/gocart2wrf/bin/gocart2wrf**
12. Optionally bogus a tropical cyclone into initial condition: Run **WRFV3/run/tc.exe**
13. Run WRF: **WRFV3/run/wrf.exe**
    a. Optionally run in one-way nest configuration: Run **WRFV3/run/wrf.exe** for outer grid, then run **WRFV3/run/ndown.exe**, then run **WRFV3/run/wrf.exe** for inner grid
    b. Optionally interpolate fine domain data back to coarse domain: Run **WRFV3/run/nup.exe**
14. Optionally generate NCAR Graphics plots of WRF netCDF output:
    a. First run **RIP4/ripdp_wrfarw** to extract variables
    b. To optionally compare two RIP data files, run **RIP4/ripcomp**
    c. To optionally cut out a subdomain of a RIP data file, run **RIP4/ripcut**
    d. To optionally interpolate data from a coarse domain to a fine domain, run **RIP4/ripinterp**
    e. To optionally interpolate data from a fine domain to a coarse domain, run **RIP4/upscale**
    f. To plot the data and optionally calculate trajectories, run **RIP4/rip**
    g. To optionally print a trajectory, run **RIP4/showtraj**
15. Optionally create GrADS binary files from WRF netCDF files: Run **ARWpost/src/ARWpost.exe**

16. Optionally convert WRF netCDF files to GRIB (required for Step 17): Run **`UPP/bin/unipost.exe`**
17. Optionally evaluate WRF forecasts against observations and/or gridded analyses:
    a. To convert PREPBUFR observation files to netCDF, run **`MET/bin/pb2nc`**
    b. To convert ASCII point observation files to netCDF, run **`MET/bin/ascii2nc`**
    c. To convert MADIS observation files to netCDF, run **`MET/bin/madis2nc`**
    d. To combine precipitation accumulations from two GRIB file and convert to netCDF, run **`MET/bin/pcp_combine`**
    e. To create a polyline masking region for verification, run **`MET/bin/gen_poly_mask`**
    f. To create ensemble statistics (mean, probability, spread, etc.) from a set of several forecast model files, run **`MET/bin/ensemble_stat`**
    g. To generate verification statistics for forecasts at observation points, run **`MET/bin/point_stat`**
    h. To generate verification statistics for a matched forecast and observation grid, run **`MET/bin/grid_stat`**
    i. To perform object-based verification, run **`MET/bin/mode`**
    j. To decompose two-dimensional forecasts and observations according to intensity and scale, run **`MET/run/wavelet_stat`**
    k. To generate statistics summary information from `point_stat`, `grid_stat`, and/or `wavelet_stat`, run **`MET/bin/stat_analysis`**
    l. To generate basic summary statistics and filtering of output from program `mode`, run **`MET/run/mode_analysis`**
    m. To regrid AFWA WWMCA products, run **`MET/bin/wwmca_regrid`**
    n. To plot AFWA WWMCA products, run **`MET/bin/wwmca_plot`**
18. Optionally evaluate land surface forecasts against observations and/or gridded analyses: Run **`LVT/make/LVT`**
19. Optionally simulate satellite data from WRF netCDF output, run **`GSDSU/QRUN_NCCS/GSDSU.x`**

g. Standard Configurations

The NU-WRF team has collectively decided on a base set of settings that capture critical variables, leverage the capabilities of the NU-WRF software and allow for improved comparisons between runs and experiments. These configurations are located in the following directories:

- `WRFV3/test/em_real/namelist.input.nuwrf.diurnal` - standard configuration for non-chemistry runs that balances day/night concerns

- `WRFV3/test/em_real/namelist.input.nuwrf.chem` – standard configuration for WRF/Chem runs. This configuration includes NU-WRF-specific variables for SOA bioemissions, Goddard microphysics-GOCART and Goddard radiation-GOCART. Note that the bioemissions is off by default as it requires additional data.

Additionally, LIS has its own independent configurations that can be used:

- `WRFV3/lis/testcases/nuwrf/lis.config.CONUS1_1KM.nuwrf` - NESDIS greenness data
- `WRFV3/lis/testcases/nuwrf/lis.config.CONUS2_1KM.nuwrf` - SPORT greenness data
- `WRFV3/lis/testcases/nuwrf/lis.config.GLOBAL_1DEG.nuwrf` - outside the CONUS

For information on NU-WRF use cases, reference the *Running NU-WRF Use Cases* documentation at:
https://modelingguru.nasa.gov/docs/DOC-1883

8. Known Issues

The following issues are known in NU-WRF:

1. *CFL errors.* Two of the supported use cases (Cases 9 and 16) randomly produce CFL errors on Discover. This behavior was first noted when WRF 3.2.1 was merged into NU-WRF, and continues through WRF 3.4.1 (the standalone community versions also exhibit this behavior). These errors have been observed with Intel 11 and 12 compilers with Intel MPI, as well as Portland Group compilers with MVAPICH2 (the latter using WRF 3.4). The cause and workaround is unknown.
2. *Namelist changes.* The "grid_fdda" setting has a new behavior introduced from WRF 3.2.1 which can cause errors. To turn off gridded fdda for a given domain, namelist variables grid_fdda, gfdda_end_h, and gfdda_interval_m must all be set to zero.
3. *Vortex tracking with WRF/Chem*. WRF/Chem will crash if it is run in vortex tracking mode. This behavior is noted in the official WRF/Chem as well. There is no known workaround.
4. *OpenMP support*. The NU-WRF build system does not explicitly support an OpenMP option. Users can select a configure option for WRFV3 that includes OpenMP, but there is no logic in place to sanity check or toggle OpenMP on or off. Also, it is not clear what effect compiling WRFV3 with OpenMP will have on WPS and UPP.

5. *GRIB2 support.* The NU-WRF build system assumes MET, UPP, and WPS are built with GRIB2 support (e.g., link to jasper, libpng, and zlib libraries). Low-level changes to the scripts may be required if a user does not wish to build with GRIB2.
6. *netCDF4 support*. The NU-WRF build system does not turn on netCDF4 compression for wrf.exe output even if the `netcdf4` build flag is used. Currently this build flag is intended to allow reading LIS netCDF4 history files with real.exe.

Appendix A. Porting NU-WRF

Currently NU-WRF is only supported on Discover and Pleiades using Intel compilers and Intel MPI. The underlying software should, however, run on other systems as long as the appropriate tools (compilers, MPI implementation, make, Perl, csh, bash, etc.) are available. Users who wish to port the system will need to take the following steps:

- Libraries:
  - Compile the libraries listed in section 5.
  - Determine the paths to the yacc binary and the flex library. Make sure yacc and flex are in your path.
  - Edit the top level build config file to update the library paths.
  - Edit the top level build config file to change modules for system binaries and libraries (compilers, MPI, etc.). If the Modules package is not installed on your system, comment out the module commands and explicitly edit the PATH and LD_LIBRARY_PATH environment variables.
- ARWpost:
  - Inspect (and if necessary edit) `ARWpost/arch/configure.defaults` to ensure a block exists for the desired operating system, hardware, and compilers. Note that ARWpost is serial only (no MPI support).
  - Run **`ARWpost/configure`** to identify the integer value of the appropriate build selection.
  - Edit the top level build config file to enter the `configure` option as environment variable ARWPOST_CONFIGURE_OPT.
  - In the top NU-WRF directory, run '**`./build.sh --config newconfig.cfg arwpost`**' to test the build.
- geos2wrf:
  - Create a new Makefile template in directory `utils/geos2wrf_2` to specify compilers and compiler flags.
  - Edit the top level build config file to enter the new makefile template name as environmental variable GEOS2WRF_MAKEFILE.
  - In the top NU-WRF directory, run '**`./build.sh --config`**

`newconfig.cfg geos2wrf`' to test the build.

- gocart2wrf:
  - Create a new Makefile template in directory `utils/gocart2wrf_2` to specify compilers and compiler flags.
  - Edit the top level build config file to enter the new makefile template name as environmental variable GOCART2WRF_MAKEFILE.
  - In the top NU-WRF directory, run '**`./build.sh --config newconfig.cfg gocart2wrf`**' to test the build.
- GSDSU:
  - Create a new Makefile template in directory `GSDSU/SRC` specifying the appropriate compilers, compiler flags, and MPI library if applicable.
  - Edit the top level build config file to enter the new makefile template name as environmental variable SDSU_MAKEFILE.
  - In the top NU-WRF directory, run '**`./build.sh --config newconfig.cfg sdsu`**' to test the build.
- lisconfig:
  - Create a new Makefile template in directory utils/lisconfig to specify compilers and compiler flags.
  - Edit the top level build config file to enter the new makefile template name as environmental variable LISCONFIG_MAKEFILE.
  - In the top NU-WRF directory, run '**`./build.sh --config newconfig.cfg lisconfig`**' to test the build.
- LVT:
  - Create a new configure.lis Makefile template in directory `LVT/arch` to specify compilers and compiler flags
  - Edit the top level build config file to enter the new makefile template name as environmental variable LVT_ARCH_CONFIGURE_FILE
  - In the top NU-WRF directory, run '**`./build.sh --config newconfig.cfg lvt`**' to test the build.
- MET:
  - Create a new Makefile template in directory `MET` specifying the appropriate compiler flags.
  - Edit the top level build config file to enter the new makefile template name as environmental variable MET_USERS_DEFS_MK.
  - In the top NU-WRF directory, run '**`./build.sh --config newconfig.cfg met`**' to test the build.
- prep_chem_sources:
  - Create a new Makefile template in directory

`utils/prep_chem_sources/bin/build` to specify compilers and compiler flags. **Note that the file name must use the naming convention `include.mk.*`.**
    - Edit the top level build config file to enter the **suffix** of the new Makefile template name as environmental variable MAKEPSC_OPT.
    - In the top NU-WRF directory, run '**`./build.sh --config newconfig.cfg prep_chem_sources`**' to test the build.
- RIP4:
    - Inspect (and if necessary edit) `RIP4/arch/configure.defaults` to ensure a block exists for the desired operating system, hardware, and compilers. Note that RIP4 is serial only (no MPI support).
    - Run **`RIP4/configure`** to identify the integer value of the appropriate build selection.
    - Edit the top level build config file to enter the `configure` option as environment variable RIP_CONFIGURE_OPT.
    - In the top NU-WRF directory, run '**`./build.sh --config newconfig.cfg rip`**' to test the build.
- UPP:
    - **NOTE: Make sure WRFV3 is ported first.**
    - Inspect (and if necessary edit) `UPP/arch/configure.defaults` to ensure a block exists for the desired operating system, hardware, compilers, and choice of parallelism. It is recommended that two blocks (for serial and MPI) exist.
    - Run **`UPP/configure`** to identify the integer value(s) of the appropriate build selection(s).
    - Edit the top level build config file to enter the `configure` options as environment variables UPP_CONFIGURE_MPI_OPT and UPP_CONFIGURE_NOMPI_OPT. Also set environmental variable UPP_USE_MPI to toggle MPI on or off.
    - Finally, in the top NU-WRF directory, run '**`./build.sh --config newconfig.cfg upp`**' to test the build.
- WPS:
    - **NOTE: Make sure WRFV3 is ported first.**
    - Inspect (and if necessary edit) `WPS/arch/configure.defaults` to ensure a block exists for the desired operating system, hardware, compilers, and choice of parallelism. It is recommended that two blocks (for serial and MPI) exist.
    - Run **`WPS/configure`** to identify the integer value(s) of the appropriate

build selection(s).
    - Edit the top level build config file to enter the `configure` options as environment variables WPS_CONFIGURE_MPI_OPT and WPS_CONFIGURE_NOMPI_OPT. Also set environmental variable WPS_USE_MPI to toggle MPI on or off.
    - In the top NU-WRF directory, run '**`./build.sh --config newconfig.cfg wps`**' to test the build.
- WRFV3:
    - Inspect (and if necessary edit) `WRFV3/arch/configure_new.defaults` to ensure a block exists for the desired operating system, hardware, compilers, and choice of parallelism. It is recommended that two blocks (for serial and MPI) exist.
    - Run **`WRFV3/configure`** to identify the integer value(s) of the appropriate build selection(s).
    - Create a new `configure.lis` makefile template in `WRFV3/lis/arch` with appropriate compiler selections. It is recommended that two templates (for serial and MPI) exist.
    - Edit the top level build config file to enter the `configure` options as environment variables WRF_CONFIGURE_MPI_OPT and WRF_CONFIGURE_NOMPI_OPT. Also list the makefile templates with environmental variables WRF_CONFIGURE_LIS_MPI and WRF_CONFIGURE_LIS_NOMPI. Also set environmental variable WRF_USE_MPI to toggle MPI on or off. Also, set environmental variable LIS_ARCH to the value appropriate for your operating system and compiler (see LIS User's Guide for options).
    - In the top NU-WRF directory, run '**`./build.sh --config newconfig.cfg wrf`**' to test the build. Also test with the 'chem' and 'kpp' targets.
- sst2wrf:
    - Create a new Makefile template in directory `utils/sst2wrf` to specify compilers and compiler flags.
    - Edit the top level build config file to enter the new makefile template name as environmental variable SST2WRF_MAKEFILE.
    - In the top NU-WRF directory, run '**`./build.sh --config newconfig.cfg sst2wrf`**' to test the build.